OPSEC MANUAL

Intro to OpSec

What is OpSec?

OpSec is short for Operational Security. Operational Security is the process of identifying mission/operation critical information that may be the target of an adversary in order to disrupt your operations. The goal is to protect and secure the information to prevent an adversary from being able to obtain it. In terms of anonymity, this can be described as a process to prevent an attacker from being able to successfully gain the information necessary to deanonymize you and/or disrupt your operations.

Why is OpSec important?

If you want to remain anonymous, you need some form of OpSec to prevent most techniques and mistakes used to deanonymize someone. Remember, there is no such thing as 100% anonymity or 100% security, the best you can do is be reasonably secure against attacks and techniques that may be used by your adversary.

Maintaining good OpSec will ensure that you remain anonymous, and your operations continue smoothly by using a structured and efficient approach to security. OpSec may look scary on paper, but it is actually very simple. There have probably been numerous times when you have actually used OpSec without even realizing it.

What is required to have good OpSec?

First you need to have a threat model, this threat model can be super simple and abstract or super simple and structured using something like Attack Trees or STRIDE, it is completely up to you. The basic requirements here are that you know what you are tying to protect and who your adversary is. From this point, you can study and learn about your adversary, including what attacks and techniques are they going to try in order to get to what you are trying to protect. From this point you can find mitigations and countermeasures in order to slow down or stop your adversary entirely, keeping what you are trying to protect safe.

In the context of anonymity, what you are trying to protect is most likely going to be your true identity. However, this may vary from person to person depending on the situation.

Basic threat modeling

What is threat modeling

Threat modeling is the process of getting to know your adversary, identifying information or attack vectors that your adversary might exploit or is capable of exploiting, and then finding a way to mitigate some of those exploits your adversary may use. When all this information is

compiled together, it is called a threat model. Everyone has a different threat model and different ways of threat modeling.

How do you develop a threat model

First, you need to know 3 things: who your adversary is, what they are capable of, and what the adversary's goal is. If you know these 3 things, you are off to a great start. You should then begin thinking of what attacks the adversary may perform in order to achieve their goal, and then think of ways to defeat those attacks. A basic example is the FBI trying to deanonymize a Tor user. You know the FBI is capable of using NITs (drive-by downloads) to deanonymize Tor users. At this point, if you think of how you would defeat a NIT, you may opt to use Tails or Whonix instead of plain Tor Browser. This is just a very basic example; different adversaries have different capabilities and techniques, often times more than one, that you have to account for. Remember that security is a moving target and is always changing, so re-evaluate your adversary and adjust your threat model accordingly from time to time.

You may also opt to use a more structured threat modeling process. This is where other threat modeling techniques come in, such as STRIDE, the CIA Triad, and Attack Trees. These also give you a basic understanding of what your adversary may try to attack, but they all require that you know who or what your adversary is.

STRIDE

The STRIDE threat modeling process was developed by Microsoft and can be applied to almost everything. STRIDE stands for, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of service, and Escalation of privileges. STRIDE is extremely thorough and effective when used correctly. STRIDE can be used as a question-and-answer process. For example the S stands for spoofing, you may ask, "Can my adversary spoof an identity?" If the answer is yes, then you may go through the process finding how your adversary would spoof an identity and how to prevent or detect identity spoofing by your adversary.

CIA Triad

The CIA triad was not invented by the Central Intelligence Agency; it actually stands for Confidentiality, Integrity, and Availability. This threat modeling process highlights the three main things you want to protect from an adversary. Confidentiality of information to prevent your adversary from knowing what it is, integrity to prevent malicious modification of information or spoofing, and availability to prevent your adversary from making information unavailable to you. For instance, on a website with a login form, you want to keep the password database confidential. It also needs integrity, so someone unauthorized can't just change the password for a user, and it needs to stay available, so a user can always log in.

Attack Trees

Attack trees are supposed to let you map out what attacks your adversary may attempt to achieve the adversary's goal and how to mitigate them effectively. You can also map out the

steps required to execute an attack in order to help mitigate that attack. An example of an attack tree is provided below.

- Adversary: FBI, Goal: De-anonymize Tor user
    - NIT (drive by download) - Mitigate by using Whonix or Tails
        - Target downloads the NIT
        - Execute the NIT
        - Gather identifying information
    - OSINT - Keep anonymous life completely separate from public life
        - Link Tor user to clear net user
        - Research clear net user

Attack trees are very flexible as you can change how you use them to fit your needs. Attack trees also help you keep track of a variety of attacks that may occur.

Choosing the right tools

You should know the limitations of different tools and what different tools are designed for. If you don't know how something works, maybe it is best not to use it. Some things like Signal are designed for privacy, while others such as Tor are designed for Anonymity. It is important to understand the difference between privacy and anonymity. Most of the tools listed here are useless if you do not use them correctly.

Tor / Tor Browser

Tor and Tor Browser are the most effective tools that when used properly can help you remain anonymous on the internet. Tor and Tor Browser's anonymity comes from the fact that all users (should) look the same and traffic is bounced around the world through relays that are run by volunteers and not by a single entity. Traffic is also encrypted when it is flowing through the Tor network. Here is where the most obvious limitation is shown, the traffic is only encrypted when it is flowing through the Tor network, not when in exits. A Tor exit node can easily see your traffic, and if you are not using HTTPS then it may be able to modify that traffic. Only use HTTPS when browsing clear net sites with Tor, this doesn't apply to onion services (sites ending in .onion like this one) as the traffic stays inside the Tor network all the way to the destination

Tor Browser is more of a target than Tor itself. Tor Browser does its best to disable dangerous web capabilities that may deanonymize a Tor user. For instance, WebRTC is practically not usable in Tor to prevent IP leaks and HTML canvas elements are randomized to prevent fingerprinting. Even though Tor Browser has disabled many dangerous web capabilities, JavaScript is not disabled by default, in fact many Tor Browser exploits come from JavaScript. JavaScript is dangerous in general, XSS exploits still exist that can steal browser cookies. Tor Browser does have 3 security levels that can be changed by clicking the shield icon on the upper right side of the browser. The safer setting is more restrictive on CSS to prevent fingerprinting attacks and also disables WebAssembly and the JavaScript JIT compiler, both of which are a large source of bugs and vulnerabilities. The safest setting disables almost

everything and only keeps the necessary things for static web pages. At this setting, JavaScript is disabled, and CSS is restricted in the same way as the Safer setting. The safest setting has the most minimal attack surface.

Over all Tor and Tor Browser can keep you relatively anonymous but when it comes to exploits and vulnerabilities, it may require tweaking the security setting or more advanced tools such as Whonix or Tails. Also note Tor cannot protect all your communications, just because you route XMPP over Tor doesn't mean that your messages are private, anonymous but not private.

XMPP with OMEMO

XMPP is a messaging protocol that on the surface looks similar to email (it's not similar at all). Different users on different servers can communicate with each other across the internet. The only problem is messages are not private, server owners and anyone in between can easily view those messages. This is where OMEMO comes in, OMEMO is an end-to-end encryption protocol designed to be used with XMPP. It is an improvement over the aging OTR encryption still commonly used in some places. OMEMO takes some ideas from the Signal protocol, OMEMO includes offline message delivery, confidentiality, deniability, integrity, authentication, and perfect forward secrecy. OMEMO however, can still be attacked if the correct precautions are not taken. A man in the middle may be able to swap OMEMO keys for their own during a key exchange, to detect and prevent this, users should verify their contact's OMEMO fingerprint through an outside channel.

OMEMO keeps your messages private when using XMPP, neither OMEMO or XMPP will keep you anonymous however they can be used in conjunction with Tor. OMEMO cannot protect your message confidentiality if an endpoint is compromised such as your contact's computer.

Whonix

Whonix is a much safer way to use Tor anonymously. Whonix uses a two VM approach, one VM for networking, one VM for browsing and other applications. This way, in order for an adversary to deanonymize you, they not only have to find a vulnerability in Tor Browser. They also have to find a vulnerability that allows them to escape the VM which is extremely difficult to do. Everything in Whonix is isolated from the rest of the machine, internet traffic is forced through Tor with no way around since the networking is in a completely separate VM. Whonix also comes with the Vanguards Tor plugin, designed to prevent guard discovery and other traffic analysis attacks that may be used to deanonymize you over a period of time.

Whonix does have its limitations however, for instance using the same Whonix-Workstation VM for different purposes or anonymous identities may allow an adversary to deanonymize you. Many users will not change the sudo password of the Whonix-Workstation VM, while this doesn't allow a VM to escape it makes it to attack. You should have multiple copies of Whonix-Workstation for different purposes, you may also opt to use the live mode for daily activities.

Tails

Tails is a live system designed to not leave a trace of anything you do on the PC the Tails USB was used on. Tails OS is a portable USB bootable operating system. Tails comes with Kleopatra for PGP via GnuPG and Tor for anonymity. The version of Tor Browser in Tails also comes with an ad blocker. The Tor Browser in Tails has also gone through some additional security hardening, mainly through the use of AppArmor. If Tor Browser is attacked with a vulnerability, AppArmor can significantly mitigate the effectiveness of that vulnerability by limiting what Tor Browser can do on the system, such as which files it can go through. Tails is also notorious for being the operating system of choice when Edward Snowden was whistleblowing on the NSA. Tails forces all traffic through Tor, traffic that refuses to go through Tor is simply dropped.

Tails however makes it obvious you are using Tails, the ad blocker in Tails' Tor Browser is unique to the Tails operating system. Tails is also not immune to vulnerabilities. Things like the email client, video player, and browser have been exploited in the past (though with great difficulty). If you are using Tails you will probably not have to worry about such exploits as the majority of them are targeted attacks. Tails works great against more generic attacks that are used like a hand grenade (like a NIT) but if you are being actively targeted, Tails will have very limited use for you Whonix would be a better choice in such scenarios.

Do note it is still recommended by many to disable JavaScript while using Tails, but if you are just browsing Reddit or doing normal generic stuff, disabling JavaScript isn't needed and would be overkill in such scenarios.

VeraCrypt

VeraCrypt is a maintained fork of the discontinued TrueCrypt. VeraCrypt can encrypt an entire device or partition or create an encrypted file container. VeraCrypt also allows you to create hidden volumes which under the correct circumstances it is impossible to prove a hidden volume exists (VeraCrypt used on an HDD along with a live system such as Tails for instance). VeraCrypt containers and volumes do not have any sort of signature and appear to consist of purely random data, making the use of encryption in some cases hard to prove. VeraCrypt is also hard to password crack due to its variable PIM and variety of hash functions, if the PIM and hash function is changed to a secret value then password cracking becomes magnitudes more difficult. VeraCrypt also allows the use of key files to make cracking even more difficult.

VeraCrypt cannot help you if you use a weak password or a password that has been used before. It is preferable to use a passphrase such as a Diceware passphrase with a length of 7-8 words, and this passphrase is only to be used with VeraCrypt. VeraCrypt also won't protect you if remnants of files remain in unencrypted space, to be sure this doesn't occur encrypt all storage including the OS or use a live system. More about this issue can be seen in VeraCrypt's documentation.

Encryption

Full Disk Encryption

Full Disk Encryption also referred to as FDE encrypts an entire storage medium, most notably Hard Disk Drives (HDDs) which is where FDE gets its name. Several forms of FDE exist today including but not limited to, BitLocker, LUKS/dm-crypt, and VeraCrypt. When a storage medium is encrypted using FDE the all the data including file metadata is encrypted, a properly encrypted drive discloses no information to an adversary other than the fact that it may be encrypted. The security of Full Disk Encryption when done correctly is decently high. Due to the nature of FDE encrypting everything, it is difficult to even infer or guess the contents of the drive without decrypting the drive. When used with a strong memorized secret such as a random Diceware passphrase, it is nearly impossible to decrypt a drive without getting the user to disclose the memorized secret or though malware such as a keylogger. Security can also be added by using a key file, without the key file the drive cannot be decrypted. Preferably, an encrypted drive should be overwritten with random data before encryption, and memorized secrets should be of sufficient length.

File Based Encryption

File encryption also referred to as File Based Encryption (FBE) is the process of individually encrypting files. There are several forms of file encryption, some with their own drawbacks and advantages. File encryption can be applied to individual files or chunks of files two notable examples of this are Picocrypt and CryFS, both work differently but both are forms of File Encryption. File encryption like Full Disk Encryption is very hard to get around when it is done correctly. However, there are some ways to infer what the encrypted file may contain. File encryption often times does not encrypt the metadata of files such as their names, sizes, and directory structures. This metadata may allow an adversary to infer what the encrypted file may be. Some file encryption schemes will account for this, such as CryFS which will hide most of the metadata.

File encryption does have its uses in areas where Full Disk Encryption is not suitable for use, such as in the cloud. File encryption also commonly includes authenticated encryption, which can detect malicious modifications to encrypted data. This is usefully for when any file is transmitted through the internet and may be intercepted by an adversary. It not only maintains confidentiality, it maintains integrity of the data. Unauthenticated encryption especially when using cipher modes such as CBC can lead to attacks such as the Padding Oracle Attack, authentication using something like HMAC or AEAD cipher prevents this attack from occurring.

Message Encryption

Message encryption such as the Signal protocol and OMEMO are designed to keep your messages secure so only you and the intended recipient can read them. These protocols make use of a variety of cryptography, including symmetric ciphers such as AES and asymmetric public key cryptography such as Elliptic Curve Cryptography(ECC) and Curve25519. As long as the keys have not been swapped out during the key exchange (can be verified by comparing fingerprints (OMEMO) or a safety number (Signal protocol)) and neither of the endpoints are compromised. The messages are almost guaranteed to not be read by anyone else but you and

your recipient. Security can further be increased by deleting messages after they have been read.

Algorithms

Choosing the right cipher is not as important as you think. Most of the time in properly made software a secure cipher is used by default such as AES or ChaCha20. However, you may want to be aware of different ciphers available to you and which ones you should definitely avoid. Most modern ciphers are not currently broken and should be OK to use. This includes the AES finalists such as Serpent, Twofish, RC6, MARS, and the AES winner Rijndael. The cipher of choice is AES (slightly modified version of Rijndael) because it has had the most analysis and testing done on it and proper implementations have practically never been broken and probably won't be broken for a very long time. AES also has hardware acceleration making it extremely fast on most processors, causing little to no performance impact on the device it is used on. ChaCha20 has now seen more use especially in TLS, it is also fast making it a great alternative to AES in some systems. Serpent, Twofish, RC6, and MARS are also good choices but shouldn't be your first choice. Like AES they are not broken but have received way less analysis and do not have any hardware acceleration so they are often times slow, however if AES is broken, these ciphers are a safe choice.

Ciphers to avoid

Here is a list of ciphers that should be avoided because they have a severe weakness or have been broken.

- RC4 - Broken stream cipher
- DES - 56 bit key can be brute forced
- Tripple DES - 64 bit block size allows birthday attacks if more than 4 GB of data is encrypted
- Blowfish - 64 bit block size allows birthday attacks if more than 4 GB of data is encrypted
- IDEA - 64 bit block size allows birthday attacks if more than 4 GB of data is encrypted
- Kuznyechik - S-Box generated with hidden algorithm*

*Kuznyechik's hidden S-Box structure has caused suspicions of a backdoor, however this was the same case for DES. The NSA had known about differential-cryptanalysis and modified the DES S-Boxes to be resistant to it, without publicly disclosing the reason. The creators of Kuznyechik could know about some new cryptanalysis technique and could have done the same to their cipher, this is more likely as the S-Box structure doesn't seem to actually weaken the cipher.

Metadata

Metadata is data about data, it can be used to identify what something is. A filename for example is metadata, it tells you what the file is or what the file contents could be. Metadata can be found everywhere, such as in photos, documents, and videos. Metadata doesn't just have to be a filename. When it comes to photos it could be the location the photo was taken, with

documents it could be the author of the document or the settings that were used to create that document.

Metadata can identify you

When sharing files, an adversary may be able to analyse them if they are shared publicly or if they are intercepted in transit. For example, if a photo contains metadata about where it was taken, an adversary can use that information to locate you. Documents contain metadata too, documents may contain metadata that can let an adversary infer what device was used to create that document or identify who the author of the document is.

Metadata is everywhere your connections to cell towers from your phone are all logged, that is metadata, and it can track you and be used for a lot of different purposes. How often you talk to someone over the internet is metadata, who you talk to is metadata, the title of your files is metadata. Metadata is hard to avoid, but it can be significantly reduced or spoofed if you manage to pay attention to how it is created or identify where it exists.

Removing metadata from files using mat2

You can remove metadata from files using a popular command line program called mat2 or if you are using Tails OS you can use the built-in Metadata Cleaner application.

Removing metadata using mat2

1. Install the mat2 package or build it from source for your Linux distribution.
2. Check a file for metadata by using executing -s filename
3. Remove the metadata by executing mat2 filename
4. A new file with the word cleaned in the name will appear, this is the file that has the metadata removed not the original file

Removing metadata using the Metadata Cleaner in Tails OS

1. Start Tails OS
2. Open the Metadata Cleaner application
3. Drag and drop a file in the Metadata Cleaner application to remove the metadata

Mobile device location tracking

Cellphones

Cellphones can be tracked even without a SIM card. Cellphones will connect to the cell tower that provides the best signal, usually this is the closest tower. Cellphones however will still contact other towers in order to estimate their signal strength. Whenever a cellphone contacts a tower, the tower takes note of the time, device, and signal strength. By using this data from multiple towers, it is trivial to pinpoint the location of a cellphone using simple maths. A

computer can automate this and perform it in less than a second, which may allow real time location tracking.

This tracking is even more accurate in urban environments where 5G compatible towers are common. Because 5G has a lower range, there has to be more towers and these towers have to be closer together, which enables more accurate tracking. Not having a 5G cellphone won't help you here because most of those towers are also 4G/LTE compatible, even if they weren't your phone would still contact them to get information about the tower. It's not the type of signal, it's the amount of towers and how close they are together that can determine how accurate location tracking can be.

When trying to defeat this kind of tracking removing the SIM card doesn't work, the phone will still contact the cell towers for emergency calls. Air-plane mode is unreliable in a lot of phones, unless your phone is running Graphene OS or an alternative focused on privacy, you shouldn't trust air-plane mode. If your phone has a removable battery, remove it, a phone with no power has no way of operating. Turning your phone off may also work but has proven to be unreliable with newer smartphones, the problem is how do you know the phone is actually off.

Possibly the only reliable solution for modern smartphones is to physically isolate the smartphone from the outside. Faraday bags, pouches, and cages provide the required isolation to prevent the phone from communicating with anything outside, this includes cell towers. Foil is a cheap alternative but depending on the phone it can take 5 all the way up to 20 layers to block all signal. If you do decide to buy a Faraday bag or pouch for your cellphone, be careful of which one you buy. Take note of how the bag closes, folding bags tend to wear out their inner signal blocking layer after a month or two of use.

Laptops

If you take a laptop with you and connect to free Wi-Fi, it is possible to track your movements by tracking which Wi-Fi networks you connect to. Most free Wi-Fi will log the MAC address of your computer, this MAC address is usually persistent across Wi-Fi networks. By using data from Wi-Fi networks, it's possible to track a device's movement by knowing what Wi-Fi networks the device has connected to. This is way less effective and accurate compared to cell phone location tracking, but it certainly something to consider.

In order to prevent this form of tracking, the MAC address of the device should be randomized. In addition, if the Wi-Fi network has a captive portal, enter different information upon every connection if the captive portal requires any information. Some operating systems like Tails OS randomize the MAC of the device by default when in use. Most Linux distributions also allow some form of MAC randomization through the usage of the network manager or through the macchanger package. If you opt to use the macchanger package, note that it sometimes doesn't work, verify the MAC has been changed before connecting to a Wi-Fi network.

Hardware solutions

Some devices geared towards privacy come with hardware switches that can disconnect things such as the microphone, camera, and wireless radios (WiFi, cell towers, and Bluetooth). These hardware switches are as effective as physically disconnecting the microphone, camera, or wireless radio. These switches are also much simpler to use and may be more reliable than a faraday bag.

Security through isolation

Why

Every piece of software has a vulnerability, eventually someone will find that vulnerability and exploit it. Isolating software from the system is an effective way to ensure that the system stays secure even if such a vulnerability is exploited. If vulnerable software is running in a VM and is then exploited, only that VM is compromised and everything else, such as the host machine and other VMs are safe. This kind of approach is very useful in situations where it is inevitable that someone is going to attempt to hack your system by exploiting vulnerable software.

Virtual Machines

Virtual Machines run operating systems and software in an isolated virtual environment. Hardware such as RAM, CPU, and GPU are also virtualized by the machine and can be set to use only a portion of the host's hardware. Virtual Machines also make use of virtual disks, usually these are hosted from a file stored on the host machine, these virtual disks help keep Virtual Machines isolated from the host. Any properly configured Virtual Machine that is running the proper settings, on the proper host, and running the proper guest operating system is going to be very difficult for things such as malware to escape.

That being said, Virtual Machines are not indestructible, malware can fill the storage of the virtual hard disk rendering it difficult to use. Virtual Machine escapes also become easier if there are misconfigurations such as enabling sharing memory with other Virtual Machines (KVM), or outdated virtualization software. Your Virtual Machines should have the absolute minimum they need to run and be used properly. If they don't need a virtual printer device, remove it. Users also shouldn't solely rely on the Virtual Machine if the OS you are running as the host is not secure, then a Virtual Machine will not protect you.

Qubes OS

Qubes OS is an operating system that virtualizes everything, the networking stack, the browser, the document editor, everything is separated in VMs called Qubes, hence the name Qubes OS. Qubes OS runs on the assumption that all software is insecure, and it will eventually be exploited. Qubes' approach to this is to isolate any compromise to minimize its effectiveness and giving the user time to react and take action, such as deleting the compromised Qube. Qubes also has Disposable Qubes, everything done inside these Qubes are deleted after use. Note that this is not for anti forensics, this is rather for situations where something that you are opening or running might be malicious, such as a malicious shell script or PDF. If that something is indeed malicious, you can just close and shutdown that Qube, and it is gone.

Qubes OS however will not protect you if you do not keep it up to date, check for updates every day. Qubes will also automatically tell you if any VMs need updating. You should also create more Qubes than just the defaults, maybe you want a dedicated Qube for your email client, you can do that, and you should do that. Using the same VM for multiple purposes defeats the purpose of isolation. Qubes OS is also particularly vulnerable to Evil Maid attacks, its Anti-Evil Maid doesn't work on TPM 2.0 and TPM 1.2 is practically non-existent any more, Qubes OS also does not support UEFI secure boot.

Whonix

Whonix is a operating system designed to be run in 2 virtual machines. Whonix comes in 2 parts, Whonix Gateway and Whonix Workstation. Whonix Workstation is where you perform all your anonymous activities and Whonix Gateway is what routes your Whonix Workstation traffic through Tor. This is another great example of security through isolation. Since the Whonix Gateway is isolated and the Whonix Workstation is isolated and Whonix Gateway is what routes the Whonix Workstation network traffic, everything goes through Tor no matter what. In order to obtain the IP address of a Whonix user, an adversary would need to escape the virtual machine/isolated environment which is extremely difficult.

For more information please check the Whonix website, Qubes OS also comes with Whonix by default.

Isolation of identities

Having multiple anonymous identities limits the amount of information an adversary can collect and significantly reduces the risk of de-anonymization. When using multiple anonymous identities, they should be kept separate, on separate partitions, operating systems, physical devices or in separate VMs. Identities should also not share information about eachother, they need to be completely separate. Identity 1 cannot say it is also Identity 2 and vice versa. The moment this isolation is lost, it defeats the purpose of having multiple anonymous identities allowing information to be linked together to further profile you and ultimately de-anonymize you.

Operations also need separation and isolation of identities. For example, if you are running a phishing campaign, the identity you use for that campaign should be different from the identity you use to post your political opinions online. Anonymous identities should also be isolated and kept completely separate from your real life identity.

There are very few cases where a single anonymous identity is required for multiple operations. An example of this is large operation, with a series of smaller operations that contribute to this larger operation. In this case you may need a single anonymous identity who's sole purpose is to oversee and manage it all. Even in scenarios like this, you will still need separate identities for performing some tasks.

Defence in depth

Defence in depth is the process of having multiple layers of security just in case one fails. Virtual Machines are an example of Defence in depth, a VM provides an extra layer of security just in case the security of what is running inside the VM fails. Multiple layers of security can slow down or even completely prevent an adversary from compromising your system. When it comes to anonymity layered security is common, Tails for instance uses AppArmor to further restrict Tor Browser just in case it is attacked. Whonix uses VMs to prevent anonymization even in the event that a vulnerability in software such as Tor Browser is exploited.

Common bad practices

Defence in depth is not about 100% security, sometimes layers are not needed. If you are trying to use the idea of defence in depth to try to defend against every possible attack under the sun, it's not going to work. An example of unnecessary layers is increasing the length of Tor circuits to 6 relays, there are people that want to do this, and it does not make you more anonymous. An attacker is more likely to attack Tor Browser itself, then do traffic analysis on the Tor network. A general rule is if adding a layer of security is going to have a minimal effect on increasing security, then you probably don't need it. Certain areas and aspects of security need more work than others and genuinely need multiple layers of security, but there are a lot of things that simply don't need it.

For instance when it comes to encryption, many focus on the encryption algorithm, some want to use cascading algorithms but in reality the keys are what need to be protected the most. If a password for encryption is used, the user should focus on using a strong password. As another layer of protection, a strong KDF such as Argon2 should be used. That is much more effective than adding more encryption algorithms. Basically what I am trying to say here is you should be careful on what areas of security you focus on. Things that are more prone to being vulnerable probably should have more layers and more attention, while other things that are established and mathematically difficult to attack don't need those layers. Don't add layers that are redundant that don't actually increase security.

Myths

Here are some of the most common myths when it comes to OpSec

Tor was/is funded by the US government and isn't secure

While the US government is one of the entities that helps fund the Tor Project, they don't do that to make Tor insecure. The US government has used Tor themselves in the past, it is not only helpful to civilians, but it is a valuable tool for government agents when they need anonymity. For instance, when they are in a hostile country and need to contact home. Tor relays are also not ran by the Tor Project, they are run by completely separate entities such as individuals who voluntarily set up relays on their home internet and non-profit organizations.

Tor isn't encrypted

This myth is usually spread by VPN companies trying to convince you to use a VPN with Tor. Tor is encrypted, in fact it uses 3 layers of encryption. When you connect to an onion site, the encryption is even more effective because the traffic stays encrypted using Tor all the way to its destination. However, most sites use HTTPS anyway, so this isn't even an issue. The point is, Tor is encrypted using strong and reliable algorithms that are properly implemented.

100% Anonymity claim

There is no way to be 100% anonymous, you can get close to 100%, but you will never get to 100% anonymity, it simply is not possible.

Telegram is an encrypted messenger

No it isn't, don't use Telegram. Telegram is not private, they do have secret chats that use encryption, but these are not turned on by default and most don't even know they exist. On top of this, Telegram rolled their own crypto, their secret chat encryption is questionable. While it is public how it works, it really isn't very good. If you want a messenger for privacy, use Signal, if you want one for privacy and anonymity use XMPP with OMEMO over Tor.

Use ProtonMail its "anonymous"

ProtonMail is not private or anonymous. They have been seen on numerous occasions giving law enforcement identifying information about specific customers. You shouldn't rely on a company or a product to keep you anonymous. ProtonMail also doesn't encrypt any metadata such as the subject lines of emails, even though it is widely supported by many email clients that support PGP.

Common Mistakes

100% security

People still believe that there is such thing as 100% security. Everything is flawed in some way, for there to be 100% security there would need to be perfect security, which simply isn't possible. If someone is determined and has enough time and funding on their hands, they will find a way around your security. A better approach is to have reasonable security and to keep up to date with the latest threats through constant software updates and applying things such as defence in depth. What this does is it makes the time and effort required not worth it or impractical for your adversary. Cryptography does this, a 128-bit key can be cracked, but the only thing stopping us is the fact it just takes too long. Anonymity also plays a role, it is harder to attack someone who you don't know anything about.

FUD

This stands for Fear Uncertainty and Disorder. Sometimes, people will spread information that is blown out of proportion or just false which creates fear, uncertainty, and disorder about a particular OpSec practice or sometimes a protocol or a piece of software. For instance, a while

ago, there was somebody who had been using LUKS on one computer and VeraCrypt on the other. Law enforcement was able to get into the LUKS encrypted computer and not the VeraCrypt encrypted computer. The person in question claimed to have been using a "secure password" and that suggested that LUKS wasn't secure, and in turn many began to believe that LUKS was not secure. The LUKS encrypted computer was actually using LUKS1 not LUKS2, which is quite easy to password crack due to its weak PBKDF2 parameters. This also means that the password that was supposed to be secure was actually not so secure because it was easier to crack. Tails in response to this decided to switch to LUKS2 and make an announcement regarding LUKS1 and LUKS2 and how it was harder and more expensive to password crack LUKS2 which used Argon2 instead of PBKDF2. If the password was longer or if it was a passphrase of sufficient length, this probably wouldn't of ever happened in the first place.

Instead of automatically believing or claiming is insecure, it is better to investigate why it didn't protect something as well as it should have, and then make changes accordingly. Sometimes things are insecure, and you should stop using them, such as 1024 bit RSA, other times something wasn't used or done properly, there are typically a lot of factors involved. Basically, don't take things at face value and do your own research.

Overlooking other vectors

When threat modelling, accounting for possible attack vectors can very difficult, you are going to overlook something. You can minimize the chance of overlooking attack vectors by studying how something you are using or trying to protect operates, the more you know the better. For instance, if you want to PGP encrypt your email, you should understand how email works. Once you know how it works, you will find more vectors such as the email metadata and the fact that PGP encryption will not protect the metadata of an email such as the subject line. You can then plan a mitigation accordingly, such as not using a subject line, putting the subject line in the body where it can be encrypted, or using a vague subject line.

http://jqibjqqagao3peozxfs53tr6aecoyvctumfsc2xqniu4xgcrksal2iqd.onion/